Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Physica D 180 (2003) 92–107
PHYSICA D
www.elsevier.com/locate/physd

# Data compression and learning in time sequences analysis

A. Puglisi [a,b,*], D. Benedetto [c], E. Caglioti [c], V. Loreto [a,b], A. Vulpiani [a,b]

[a] *Physics Department, "La Sapienza" University in Rome, Piazzale Aldo Moro 5, 00185 Rome, Italy*
[b] *INFM, Center for Statistical Mechanics and Complexity, Rome, Italy*
[c] *Mathematics Department, "La Sapienza" University in Rome, Piazzale Aldo Moro 2, 00198 Rome, Italy*

Received 22 June 2002; received in revised form 10 December 2002; accepted 27 January 2003
Communicated by M. Vergassola

**Abstract**

Motivated by the problem of the definition of a distance between two sequences of characters, we investigate the so-called learning process of a typical sequential data compression schemes. We focus on the problem of how a compression algorithm optimizes its features at the interface between two different sequences $A$ and $B$ while zipping the sequence $A + B$ obtained by simply appending $B$ after $A$. We show the existence of a scaling function (the "learning function") which rules the way in which the compression algorithm learns a sequence $B$ after having compressed a sequence $A$. In particular it turns out that there exists a cross-over length for the sequence $B$, which depends on the relative entropy between $A$ and $B$, below which the compression algorithm does not learn the sequence $B$ (measuring in this way the cross-entropy between $A$ and $B$) and above which it starts learning $B$, i.e. optimizing the compression using the specific features of $B$. We check the scaling on three main classes of systems: Bernoulli schemes, Markovian sequences and the symbolic dynamic generated by a nontrivial chaotic system (the Lozi map). As a last application of the method we present the results of a recognition experiment, namely recognize which dynamical systems produced a given time sequence. We finally point out the potentiality of these results for segmentation purposes, i.e. the identification of homogeneous sub-sequences in heterogeneous sequences (with applications in various fields from genetic to time-series analysis).


*Keywords:* Time sequence analysis; Compression algorithm; Characterization of complexity

## 1. Introduction

The modern approach to time-series analysis based on the theory of dynamical systems and information theory (IT) has represented a major advance in the description and comprehension of a wide range of phenomena, from geophysics to industrial processes [1,2]. Time series represent a particular example of the wider category of strings of characters which also includes as further examples texts or genetic sequences (DNA, proteins). When analyzing a string of characters the main question is to extract the information it brings. For example, in a DNA sequence this would correspond to the identification of the sub-sequences codifying the genes and their specific functions. On the

* Corresponding author. Present address: Physics Department, "La Sapienza" University in Rome, Piazzale Aldo Moro 5, 00185 Rome, Italy. Tel.: +39-0649913459; fax: +39-064463158.
*E-mail address:* andrea.puglisi@roma1.infn.it (A. Puglisi).

0167-2789/03/$ – see front matter 
doi:10.1016/S0167-2789(03)00047-2

other hand for a written text one could be interested in recognizing the language in which the text is written, the subject treated or its author. For time series one could be interested in the extraction of specific features or trends [3].

In the spirit of having specific tools for the measurements of the amount of information brought by a sequence, it is rather natural to approach the problem from a very interesting point of view: that of IT [4,5]. Born in the context of electric communications, IT theory has acquired, since the seminal paper of Shannon [4], a leading role in many other fields as computer science, cryptography, biology and physics [5]. In this context the word information acquires a very precise meaning, namely that of the entropy of the string, a measure of the *surprise* the source emitting the sequences can reserve to us.

It is important to stress that IT deals with ensembles of sequences emitted by an ergodic source, while one is typically forced to treat a single sequence. In this spirit an appropriate concept is that of algorithmic complexity (AC) [6–9]. The AC (sometimes called also Kolmogorov complexity) of a string of characters is given by the length (in bits) of the smallest program which produces as output the string. A string is said to be complex if its complexity is proportional to its length. This definition is really abstract, in particular it is impossible, even in principle, to find such a program [10]. Since this definition tells nothing about the time the best program should take to reproduce the sequence, one can never be sure that somewhere else there does not exist another shorter program that will eventually produce the string as output in a larger (eventually infinite) time; this impossibility is related to the Turing's theorem on the halting problem and to the Gödel's theorem [10].

Despite the impossibility to compute the AC of a sequence, one has to recall that there are algorithms explicitly conceived to give a good approximation to the AC [10]. Since the AC of a string fixes the minimum number of bits one should use to reproduce it (optimal coding), it is intuitive that a typical zipper, besides trying to reduce the space occupied on a memory storage device, can be considered as an entropy meter. The better will be the compression algorithm, the closer will be the length of the zipped file to the optimal coding limit and the better will be the estimate of the AC provided by the zipper.

It is well known that compression algorithms represent a powerful tool for the estimation of the AC or more sophisticated measures of complexity [11–13] and several applications have been drawn in several fields [14] from dynamical systems theory (the connections between IT and dynamical systems theory are very strong and go back all the way to the work of Kolmogorov and Sinai; for a recent overview see [15–17]) to linguistics (an incomplete list would include [18–25]) and genetics (see [26–28], and references therein).

Some of us have recently proposed a method [25] for context recognition and context classification of strings of characters or other equivalent coded information. The remoteness between two sequences $A$ and $B$ was estimated by zipping a sequence $A + B$ obtained by appending the sequence $B$ after the sequence $A$ and using the *gzip* compressor [29] (whose core is provided by the Lempel–Ziv 77 (LZ77) algorithm [30]). This idea is used for authorship attribution and, defining a suitable distance between sequences, for languages phylogenesis.

The idea of appending two files and zip the resulting file in order to measure the remoteness between them had been previously proposed by Loewenstern et al. [28] (using *zdiff* routines) who applied it to the analysis of DNA sequences, and by Khmelev and coworkers [23] who applied the method to authorship attribution. In particular here the method is extensively tested using many different zippers, including *gzip*. Though the idea is the same the practical implementation differs from the one proposed in [25].

In this paper we extend the analysis of [25] by considering more in detail the features of data compression algorithms when applied to generic strings of characters. The specific question we raise here is how LZ77-like compression algorithms behave at the interface between two different files. More specifically we shall focus on the process by which a typical zipper *learns* the sequence it is processing and how it uses previous information acquired while zipping a given file to zip a second different file. We point out in particular the existence of a scaling function which rules the way in which the compression algorithm learns the sequence $B$ after having zipped sequence $A$. These kind of problems are closely related to the so-called segmentation problem, i.e. the identification

of homogeneous sub-sequences in heterogeneous sequences (with applications in various fields from genetic to time-series analysis).

Since in this case we are interested in exploring the features of the compression algorithms we shall use as benchmark systems time sequences issued by dynamical systems of increasing complexity. In particular the scaling function is checked numerically for three main classes of systems: Bernoulli schemes, Markovian sequences and the nontrivial symbolic dynamic generated by the so-called Lozi map. As a last application of the method we present the results of a recognition experiment, namely recognize which dynamical systems produced a given time sequence.

The outline of the paper is as follows. In Section 2 we recall some basic definitions. Section 3 is devoted to the discussion of data compression techniques as well as to recall the definition of relative entropy and the Ziv and Merhav algorithm [11] for its measure. In Section 4 we study what happens when applying the LZ77 [30] algorithm to a sequence obtained appending two different sequences. In Section 5 we analyze numerically the results of Section 4. In Section 6 we perform a recognition experiment on sequences generated by the Lozi map. Finally in Section 7 we draw the conclusions and discuss possible fields of application for these techniques.

## 2. Basic concepts

Originally IT was introduced by Shannon [4] in the practical context of electric communications. The powerful concepts and techniques of IT allow for a systematic study of sources emitting sequences of discrete symbols (e.g. binary digit sequences) and in the last decades there have been shown the deep relations between IT and other fields as computer science, cryptography, biology and chaotic systems [5,17].

Consider a symbolic sequence $\sigma_1\sigma_2\cdots$, where $\sigma_t$ is the symbol emitted at time $t$ and each $\sigma_t$ can assume one of $m$ different values. Assuming that the sequence is stationary we introduce the $N$-block entropy:

$$H_N = -\sum_{\{C_N\}} p(C_N)\ \ln\ p(C_N), \tag{1}$$

where $p(C_N)$ is the probability of the $N$-word $C_N = (\sigma_t\sigma_{t+1}\cdots\sigma_{t+n-1})$, and $\ln\ = \log_e$. The differential entropies:

$$h_N = H_{N+1} - H_N \tag{2}$$

have a rather obvious meaning; $h_N$ is the average information supplied by the $(N+1)$th symbol, provided the $N$ previous ones are known. Noting that the knowledge of a longer past history cannot increase the uncertainty on the next outcome, one has that $h_N$ cannot increase with $N$, i.e. $h_{N+1} \le h_N$. Now we are ready to introduce the Shannon entropy for an ergodic stationary process:

$$h = \lim_{N\to\infty} h_N = \lim_{N\to\infty} \frac{H_N}{N}. \tag{3}$$

It is easy to see that for a $k$th order Markov process, i.e. such that the conditional probability to have a given symbol only depends on the last $k$ symbols, $p(\sigma_t|\sigma_{t-1}\sigma_{t-2},\dots) = p(\sigma_t|\sigma_{t-1}\sigma_{t-2},\dots,\sigma_{t-k})$, then $h_N = h$ for $N \ge k$.

The Shannon entropy $h$ measures the average amount of information per symbol and it is an estimate of the "surprise" the source emitting the sequence reserves to us. The fact is remarkable that, under rather natural assumptions, the entropy $H_N$ apart from a multiplicative factor, is the unique quantity which characterizes the "surprise" of the $N$-words [31]. Let us try to explain in which sense entropy can be considered as a measure of a surprise. Suppose that the surprise one feels upon learning that an event E has occurred depends only on the probability of E. If the event occurs with probability 1 (sure!) our surprise in its occurring will be zero. On the other hand if the probability of occurrence of the event E is quite small our surprise will be proportionally large. For a single event occurring with probability $p$ the surprise is proportional to $-\ln\ p$. Let us consider now a random variable $X$, which

can take $N$ possible values $x_1, \ldots, x_N$ with probabilities $p_1, \ldots, p_N$, the expected amount of surprise we shall receive upon learning the value of $X$ is given precisely by the entropy of the source emitting the random variable $X$, i.e. $-\sum p_i \ln p_i$.

A theorem, due to Shannon and McMillan [4,31], expresses in a precise way how $h$ quantifies the "complexity" of the source: if $N$ is large enough, the set of $N$-words $\{C_N\}$ can be partitioned in two classes, $\Omega_1(N)$ and $\Omega_2(N)$ such that all the words $C_N \in \Omega_1(N)$ have probability $p(C_N) \sim \mathrm{e}^{-hN}$ and

$$\sum_{C_N \in \Omega_1(N)} p(C_N) \to 1 \quad \text{for } N \to \infty, \tag{4a}$$

$$\sum_{C_N \in \Omega_2(N)} p(C_N) \to 0 \quad \text{for } N \to \infty. \tag{4b}$$

An important implication of the theorem is that the number of typical sequences $\mathcal{N}_{\mathrm{eff}}(N)$ (those in $\Omega_1(N)$) effectively observable is

$$\mathcal{N}_{\mathrm{eff}}(N) \sim \mathrm{e}^{hN}. \tag{5}$$

Note that in nontrivial cases, in which $h < \ln m$, $\mathcal{N}_{\mathrm{eff}}(N) \ll m^N$, $m^N$ being the total number of possible $N$-words. Let us remark that the Shannon–McMillan theorem for processes without memory is nothing but the law of large numbers. Writing Eq. (5) in the form $H_N \sim \ln \mathcal{N}_{\mathrm{eff}}$ one can understand its relation with the Boltzmann equation in statistical thermodynamics $S \propto \ln W$, $W$ being the number of possible microscopic states and $S$ the thermodynamic entropy.

An important result is the relation between the maximum compression rate of a sequence $(\sigma_1\sigma_2\cdots)$ expressed in an alphabet with $m$ symbols, and $h$. If the length $T$ of the sequence is large enough, then it is not possible to compress it into another sequence (with an alphabet with $M$ symbols) whose size is smaller than $Th/\ln M$. Therefore, noting that the number of bits needed for a symbol in an alphabet with $M$ symbol is $\ln M$, one has that the maximum allowed compression rate is $h/\ln M$. Perhaps the simplest way to compress, at least at a conceptual level, is via the Shannon–Fano procedure which is able to reach asymptotically the maximum allowed compression rate [32]. Also the popular Lempel–Ziv coding [30] (see in the following for a short discussion) gives the same asymptotic results.

We stress the fact that $h$ is an asymptotic quantity which gives the behavior of $H_N$ (or equivalently $h_N$) at large $N$, i.e. $h \simeq H_N/N$ for $N \gg 1$. On the other hand the features of $H_N$ (or $h_N$) for moderate $N$ are rather important in all nontrivial processes (i.e. with memory). An important quantity introduced to measure these effects and characterize the properties of a sequence from the behavior of $H_N$, is the so-called *excess entropy* [33] or *effective measure complexity* [34] (for a recent overview and other references where these concepts have been discussed see [35]). Let us introduce

$$\delta h_N = h_{N-1} - h_N \tag{6}$$

and the excess entropy (or effective measure complexity) $C$ as

$$C = \sum_{N=1}^{\infty} N\delta h_N. \tag{7}$$

It is not difficult to realize that, for large $N$, one has

$$H_N \simeq C + hN. \tag{8}$$

In trivial processes (e.g. Bernoulli schemes), $C = 0$, on the other hand $C$ can be nonzero in cases with zero $h$ (e.g. periodic sequences). Particularly interesting are the cases where $h$ is positive and $C$ is not negligible; nontrivial

examples are given by dynamical systems producing sequences with memory and forbidden words, such as the Lozi map, which is discussed in Section 5.3.

## 3. Data compression and complexity

As already mentioned there exists an important relation between the maximum compression rate achievable for a given sequence and its AC. We have as well stressed that AC, at variance with IT, does not deal with an ensemble of sequences, but with a single sequence. On the other hand there is a rather important relation between the Kolmogorov complexity (or AC) $K_N(W_N)$ of a $N$-word $W_N$ and $H_N$:

$$\frac{1}{N}\langle K_N\rangle = \frac{1}{N}\sum_{W_N} K_N(W_N)P(W_N) \underset{N\to\infty}{\to} \frac{h}{\ln 2}, \tag{9}$$

where $K_N$ is the binary length of the shorter program needed to specify the $N$-word $W_N$.

In Section 1 we have already outlined that, despite the impossibility to compute the AC of a sequence, data compression techniques represent effective tools for an estimation of AC or other measures of complexity. In particular any such algorithm provides with an upper bound of the real AC.

A great improvement in the field of data compression has been represented by the Lempel and Ziv algorithm (LZ77) [30] (used, for instance, by *gzip* and *zip*). It is interesting to briefly recall how it works. Let $x = x_1, \ldots, x_N$, be the sequence to be zipped. The LZ77 algorithm proceeds sequentially along the sequence. Let us suppose that the first $n$ characters have been codified. Then the zipper looks for the largest integer $m$ such that the string $x_{n+1}, \ldots, x_{n+m}$ already appeared in $x_1, \ldots, x_n$. Then it codifies the string found with a two-number code composed by: the distance between the two strings and the length $m$ of the string found. If the zipper does not find any match then it codifies the first character to be zipped, $x_{n+1}$, with its name. This eventuality happens, for instance, when codifying the first characters of the sequence, but this event becomes very unfrequent as the zipping procedure goes on.

LZ77 algorithm has the following remarkable property: if it encodes a text of length $L$ emitted by an ergodic source (precisely a typical sequence emitted by a stationary stochastic process with finite memory) whose entropy per character is $h$, then the length of the zipped file divided by the length of the original file tends to $h/\ln 2$ when the length of the text tends to $\infty$. In other words it does not encode the file in the best way but it does it better and better as the length of the file increases. More precisely the code rate, i.e. the average number of bits per symbol needed to encode the sequence, can be written as

$$\text{code rate} = \frac{\text{average number of bits to encode the phrase}}{\text{length of the phrase}} \simeq \frac{\ln N + \ln L_N + \mathrm{O}(\ln \ln L_N)}{L_N \ln 2}, \tag{10}$$

where $L_N$ is the average length of the phrase substituted and $N$ the length of the part of the sequence already analyzed. Note that $\ln N$ is the number of bits needed to encode the part of the pointer describing the distance, while $\ln L_N$ is the number of bits needed to encode the part of the pointer describing the length of the substitution. Recalling [36] that for $N \to \infty$ one has that $L_N \to \ln N/h$ (in probability) one obtains

$$\text{code rate} \simeq \frac{h}{\ln 2} + \mathcal{O}\left(\frac{\ln \ln N}{\ln N}\right), \tag{11}$$

i.e. the LZ77 algorithm converges asymptotically to the Shannon entropy even though the convergence is extremely slow. It is important to remind that the redundancy of the LZ77 coding has been rigorously determined by Savari [37].

The first conclusion one can draw is therefore about the practical possibility to measure the entropy of a large enough sequence simply by zipping it. For example, if one compresses an English text the length of the zipped file is

typically of the order of one-fourth of the length of the initial file. An English file is encoded with 1 byte (8 bits) per character. This means that after the compression the file is encoded with about 2 bits per character. Obviously this is not yet optimal. Shannon with an ingenious experiment showed that the entropy of the English text is something between 0.6 and 1.3 bits per character [38] (for a recent study see [39]).

### 3.1. Relative entropy

Another important quantity we need to recall is the notion of relative entropy or Kullback–Leibler divergence [40–42] which is a measure of the statistical remoteness between two distributions. Its essence can be easily grasped with the following example. Let us consider two ergodic sources $A$ and $B$ emitting sequences of independent 0 and 1: $A$ emits a 0 with probability $p_A$ and 1 with probability $1 - p_A$, while $B$ emits 0 with probability $p_B$ and 1 with probability $1 - p_B$. As already described, the compression algorithm applied to a sequence emitted by $A$ will be able to encode the sequence almost optimally, i.e. with an average number of bits per character equal to $-p_A \ln p_A - (1 - p_A) \ln (1 - p_A)$. This optimal coding will not be the optimal one for the sequence emitted by $B$. In particular the entropy per character of the sequence emitted by $B$ in the coding optimal for $A$ will be the cross-entropy per character:

$$\tilde{h}(B\|A) \equiv \tilde{h}(p_B\|p_A) = -p_B \ln p_A - (1 - p_B) \ln (1 - p_A), \tag{12}$$

while the entropy per character of the sequence emitted by $B$ in its optimal coding is $-p_B \ln p_B - (1 - p_B) \ln (1 - p_B)$. The number of bits per character wasted to encode the sequence emitted by $B$ with the coding optimal for $A$ is the relative entropy per character of $A$ and $B$:

$$d(B\|A) \equiv d(p_B\|p_A) = -p_B \ln \frac{p_A}{p_B} - (1 - p_B) \ln \frac{1 - p_A}{1 - p_B}. \tag{13}$$

A linguistic example will help to clarify the situation: transmitting an Italian text with a Morse code optimized for English will result in the need of transmitting an extra number of bits with respect to another coding optimized for Italian; the difference is a measure of the relative entropy.

Given two stationary and ergodic sources of symbols of a same alphabet, of measure $p_A$ and $p_B$, using the notation of Eq. (1), the $N$-block cross-entropy is defined as

$$\tilde{H}_N(B\|A) = -\sum_{\{C_N\}} p_B(C_N) \ln p_A(C_N), \tag{14}$$

while the $N$-block relative entropy is

$$D_N(B\|A) = -\sum_{\{C_N\}} p_B(C_N) \ln \frac{p_A(C_N)}{p_B(C_N)} = \tilde{H}_N(B\|A) - H_N(B), \tag{15}$$

where $H_N(B)$ is the $N$-block entropy of the source $B$. The cross-entropy per character and the relative entropy for character are defined as follows:

$$\tilde{h}(B\|A) = \lim_{N\to\infty} \frac{1}{N} \tilde{H}_N(B\|A) = -\lim_{N\to\infty} \frac{1}{N} \sum_{\{C_N\}} p_B(C_N) \ln p_A(C_N) \tag{16}$$

and

$$d(B\|A) = \lim_{N\to\infty} \frac{1}{N} D_N(B\|A) = -\lim_{N\to\infty} \frac{1}{N} \sum_{\{C_N\}} p_B(C_N) \ln \frac{p_A(C_N)}{p_B(C_N)} = \tilde{h}(B\|A) - h(B), \tag{17}$$

where $h(B)$ is the entropy per character of the source $B$.

Let us stress that in general all these quantities could be infinite simply because of sequences emitted by the first source and not existing in the second (i.e. $p_A(C_N) = 0$ and $p_B(C_N) \neq 0$ for some sequences $C_N$). In Section 5 we will discuss how to treat this problem in practical applications.

Finally we mention that recently an algorithm has been proposed by Ziv and Merhav [11] for the measurement of the relative entropy. The method is based on a procedure very similar to the one used in the LZ77.

## 4. Relative entropy and learning

Let us now describe how the LZ77 algorithm zips a file obtained by appending a file $B$ of length $L_B$ to a file $A$ of length $L_A$. The files $A$ and $B$ are emitted by two ergodic sources with ergodic measures given by $p_A$ and $p_B$, respectively. We will use the symbols $A$ and $B$ to denote indifferently the files and their sources.

In particular it is important to understand how the second file is encoded once the sequential zipper starts reading it. Very roughly what happens is the following. First of all the zipper encodes file $A$. Then it begins encoding file $B$. Initially the zipper will find the longest match of the file $B$ in the file $A$. After a while, however, the longer is the fraction of $B$ already analyzed, the larger will be the probability to find the longest match in file $B$ itself. Asymptotically the longest matches of file $B$ will be found only inside $B$. This means that we can roughly describe this process as a two step process: in a first time the zipper tends to optimize the coding for the $A$ part while in a second time it encodes the $B$ file with the coding obtained for the $A$ part (transient) as well as with the statistics proper of the $B$ file (which will asymptotically dominate). For these reasons the zipping procedure of $A + B$ can be seen as a sort of learning process.

It is convenient here to consider the following idealized problem. Let $\sigma = (\sigma_1\sigma_2\cdots)$ be an infinite sequence extracted with measure $p_B$. Let $\sigma_A$ be a sequence of length $L_A$ extracted with the measure $p_A$, and $\sigma_B$ a sequence of length $L_B$ extracted with the measure $p_B$. Let $n_A, n_B$ be the largest integers $m$ such that $(\sigma_1\sigma_2\cdots\sigma_m)$ is contained in $\sigma_A, \sigma_B$, respectively. Let us define the function $P(L_A, L_B)$ as the probability that $n_A > n_B$. In the zipping procedure $P(L_A, L_B)$ will be the probability that, once the zipper is scanning the $B$ part of the $A + B$ file, it finds a matching in the $A$ part rather than in the $B$ part.

We can say that the typical distance between two occurrences of the same substring is inversely proportional to the probability of the substring itself. An argument based on the Shannon–McMillan theorem [11] shows that the probability of occurrence of a string of length $N$ of the sequence $\sigma$ with respect to the measure $p_A$ is asymptotically given by $\mathrm{e}^{-N[h(B)+d(B\|A)]}$.

Therefore the length $n_A$ of the longest match found in $A$ will be obtained approximately by imposing $L_A\,\mathrm{e}^{-N[h(B)+d(B\|A)]} = 1$, whose inversion gives

$$n_A = \frac{\ln L_A}{h(B) + d(B\|A)}. \tag{18}$$

Analogously the length of the longest match found in the part of file $B$ already encoded will be given approximately by

$$n_B = \frac{\ln L_B}{h(B)}. \tag{19}$$

Therefore we expect that if

$$\frac{\ln L_B}{h(B)} \ll \frac{\ln L_A}{h(B) + d(B\|A)}, \tag{20}$$

the longest match will be found in $A$, i.e. $P(L_A, L_B) \simeq 1$, while if

$$\frac{\ln L_B}{h(B)} \gg \frac{\ln L_A}{h(B) + d(B\|A)}, \tag{21}$$

one expects to find it in $B$, i.e. $P(L_A, L_B) \simeq 0$. These relations allow for defining a cross-over length for the sequence $B$ given by

$$L_B^* \simeq L_A^\alpha \tag{22}$$

with $\alpha = h(B)/(h(B) + d(B\|A))$. This is the length below which the compression algorithm does not learn the sequence $B$ (measuring in this way the cross-entropy between $A$ and $B$) and above which it learns $B$, i.e. optimizes the compression using the specific features of $B$.

It is important now to focus more precisely on the transient region where, as already noticed, there takes place a sort of learning process. In order to do this we first consider the case in which the two sequences $A$ and $B$ are (0, 1) Bernoulli sequences of length $L_A$ and $L_B$, respectively. Afterward we shall try to generalize the result.

The first source emits 0 with probability $p_A$ and 1 with probability $1 - p_A$. The second source emits 0 with probability $p_B$ and 1 with probability $1 - p_B$. Therefore, in a typical sequence of length $N$ emitted by the second source, 0 will appear approximately $p_B N$ times while 1 will appear approximately $(1 - p_B)N$ times. More precisely we can say that $m_0$ (the number of zeros in the second sequence) is approximately a Gaussian random variable with average $p_B N$ and variance $\mathrm{O}(N)$.

By neglecting the fluctuations of $m_0$, one has that the probability of this sequence with respect to the measure of the first source will be approximately given by

$$p_A^{p_B N}(1 - p_A)^{(1-p_B)N} = \mathrm{e}^{N[p_B \ln p_A + (1-p_B) \ln (1-p_A)]}. \tag{23}$$

This expression is nothing but $\mathrm{e}^{-N[h(B)+d(B\|A)]}$.

Now let us take into account the fluctuations. $m_0$ has random fluctuations of order $\sqrt{N}$ around its average. This fluctuations induce fluctuations of the probability of this string with respect to the measure $p_B$. We then expect fluctuations of order $\sqrt{n_A} = \mathrm{O}(\sqrt{\ln L_A})$ of the length $n_A$ of the longest match found in the first string. The same is true for $n_B$. It seems therefore reasonable that the distributions of $n_A$, $n_B$ tend to Gaussian distributions with averages given by (18) and (19), and variances given by $c_A \ln L_A$, and $c_B \ln L_B$ where $c_A$ and $c_B$ are constants.

Therefore $P(L_A, L_B)$ is the probability that a Gaussian variable is larger than another Gaussian variable. This problem can be easily analyzed and lead us to conjecture that $P(L_A, L_B)$ converges to a function when suitably scaled: more precisely

$$P(x, y) \underset{x,y\to\infty}{\to} f\left(\frac{\ln x - \alpha \ln y}{\sqrt{\ln x + \ln y}}\right). \tag{24}$$

On the basis of large deviations theory [43], we expect this conjecture to be valid for sequences with short term memory, i.e. where the correlations decay sufficiently fast. In the next section we shall numerically check this conjecture.

Let us conclude this section by noticing that the fluctuations of the string found by LZ77 (the fluctuations of $n_B$ in the case analyzed here), have been characterized in [44] in the case of a Markovian source. In particular it has been proved that the length of the longest phrase found is asymptotically distributed with a Gaussian distribution with average $\ln L/h$, and variance $\propto \ln L$, where $h$ is the entropy of the source. Other very interesting related problems have been considered in [45,46].

## 5. Numerical results

The hypothesis for the scaling form (24) introduced in the previous section for the so-called learning function, can be tested for finite size sequences generated according to some stochastic rule, e.g. with pseudo-random number

generators or with some nontrivial dynamical systems. In this section we shall check this hypothesis in three cases featuring an increasing *complexity*: Bernoulli schemes, Markov processes and finally non-Markovian processes obtained with an empirical symbolic sequence generated by the Lozi map.

### 5.1. Bernoulli scheme

The simplest random sequence of symbols is generated by a Bernoulli scheme: at each time $t$ the symbol $\sigma_t$ is 0 with probability $p$ and 1 with probability $1-p$, with $p \in [0,1]$. This is the sequence of biased (unfair) coin tosses; it is very easy to see that $h = h_n = H_n/n = -[p \ln p + (1-p) \ln (1-p)]$ for every $n \geq 1$, and the effective measure complexity is $C = 0$.

We have generated a sequence $A$ of 0's and 1's of length $L_A$ with a probability $p_A$ for 0's, and then a set of 5000 sequences $B$ of length $L_B$ where 0's occur with probability $p_B$. For these cases the relative entropy per character is given by Eq. (13). For each sequence of this set, the following numerical experiment has been performed:

1. A sequence $AB$ (of length $L_A + L_B$) is obtained appending the $B$ sequence to the end of the $A$ sequence.
2. One starts scanning the sequence $AB$ from the point $i = i_{\text{start}} = L_A + 1$, i.e. from the first character of the sequence $B$.
3. One looks for the longest sub-sequence that:
   (a) starts at $i$;
   (b) is identical to a sub-sequence contained in the part $[1, i]$ of the joint sequence $AB$.
   The length of this maximum sub-sequence is called $n_{\max}$.
4. The index $i$ is increased by $n_{\max}$. If $i < L_A + L_B$ the algorithm goes to 3, otherwise the algorithm stops.

In the above procedure, one keeps track of the statistics of the sub-sequence matchings; in particular we are interested in the number of sub-sequences found in $A$ or in $B$ as a function of $L_B$. At the beginning of the scanning procedure most of the matchings are found in $A$. When $L_B$ is large enough, sub-sequence matchings found in $B$ can be competitive with their length against the ones found in $A$. The procedure of averaging over many "realizations" of sequence $B$ allows for a smooth statistics, i.e. a smooth curve $P(L_A, L_B)$ versus $L_B$ with fixed $L_A$.

Fig. 1 reports the curves obtained with the above procedure for $P(L_A, L_B)$ versus $L_B$ for different values of $L_A$ and different choices of the pair $(p_A, p_B)$, as well as their collapse using the scaling function (24). The collapse is indeed very satisfying, bringing the first evidence for the conjecture in (24). In the picture is also shown the failure of the scaling form when $\alpha$ is too small (*pluses* and *crosses* in the inset, not reported in the main plot). This happens when the two sequences are too different or when the second sequence has an entropy $h$ very low; in both cases the convenience of parsing the sub-sequences of $B$ with sub-sequences of its own past (and not from $A$) comes too early, as can be seen in the inset of the figure. As a consequence of this, the length of $A$ does not matter for the parsing of sequence $B$ and the two curves obtained with different $L_A$ (those with $\alpha = 0.156$) are identical.

### 5.2. Markovian sequences

The natural step after Bernoulli schemes, is a test using sequences generated by means of Markov chains. A Markov chain is a random process with discrete states, where the probability of every state is determined by one or more previous states. The order of Markov chains is the number of previous states influencing the present, e.g. for a Markov chain of order $k = 1$ the probability of having a certain symbol depends only on the previous symbol and is determined by its conditional probability $W_{ij} = P(\sigma_t = j | \sigma_{t-1} = i)$. We have tested the scaling hypothesis on

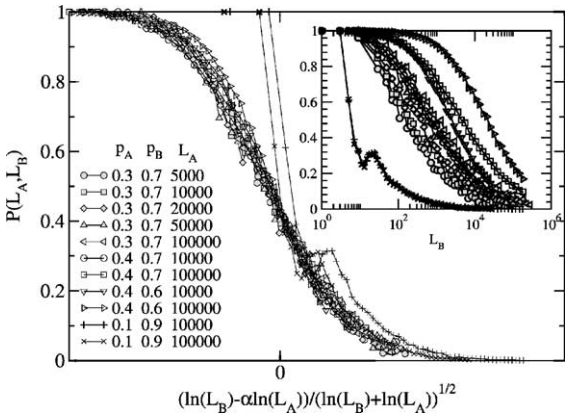


Fig. 1. Collapse of $P(L_A, L_B)$ versus the rescaled coordinate (discussed in the text), for different pairs of Bernoulli processes with different probabilities of symbol "zero" $(p_A, p_B)$ and with different lengths $L_A$ of buffer $A$. In the inset the same data are shown versus $L_B$, i.e. without any rescaling. The values of $\alpha$ are the following: 0.643 for $(p_A, p_B) = (0.3, 0.7)$, 0.768 for $(p_A, p_B) = (0.4, 0.7)$, 0.892 for $(p_A, p_B) = (0.4, 0.6)$, 0.156 for $(p_A, p_B) = (0.1, 0.9)$.

the Lempel–Ziv parsing procedure of pairs of two symbols, order one, symmetric Markov chains. This means that both $A$ and $B$ are sequences of 0's and 1's and that their transition matrix is of the form:

$$W = \begin{pmatrix} w & 1-w \\ 1-w & w \end{pmatrix} \tag{25}$$

with $w \in [0, 1]$ the probability of repeating the previous symbol. The sequences $A$ and $B$ have different transition matrices, i.e. $w = w_A$ for $A$ and $w = w_B$ for $B$. In practice a sequence obtained with $w$ near 1 is something like 11111100000011111100000 ..., while a sequence obtained with $w$ near 0 is like 010101001010101101010 ....

For a Markov chain of order 1 one has $H_N = H_1 + (N-1)h$ and $C = H_1 - h$. Moreover we are interested in the cross-entropy per character $\tilde{h}_N = \tilde{H}_{N+1}(B\|A) - \tilde{H}_N(B\|A)$ versus $N$, where, following the notation of Eq. (14)

$$\tilde{H}_N = -\sum_{\{C_N\}^*} p_B(C_N) \ln p_A(C_N) = H_N(B) + D_N^*(B\|A), \tag{26}$$

where $\{C_N\}^*$ is the set of $N$-sequences contained both in $A$ and $B$. In formula (26) $D_N^*(B\|A)$ is given by the definition (15) with the restriction that the sum runs only on the $N$-sequences contained in both $A$ and $B$. This defines, coherently with (17), the limit $d^*(B\|A) = \lim_{N\to\infty} 1/N D_N^*(B\|A)$. If we consider infinite sequences $A$ and $B$ and a two states Markov process (as the one introduced in this section) then $\{C_N\}^* \equiv \{C_N\}$, i.e. the whole set of $2^N$ sequences of length $N$ is explored by both dynamics and therefore $D_N^* \equiv D_N$ and $d^* \equiv d$. For the kind of Markov chain described by the transition matrix in (25), we can therefore calculate

$$\tilde{H}_N = \tilde{H}_1 - (N-1) \sum_{\{S_i S_j\}} P_B(S_i) W_{ij}^B \ln W_{ij}^A, \tag{27a}$$

$$\tilde{h}_N = \tilde{h}(B\|A) = h(A) + d(B\|A) = -\sum_{\{S_i S_j\}} P_B(S_i) W_{ij}^B \ln W_{ij}^A. \tag{27b}$$

More in general the above formulas hold if $W_{ij}^A$ is positive when $W_{ij}^B$ is positive.

In Fig. 2 we show the effects of finiteness of the sequences $A$ and $B$ on $h_N$ and $\tilde{h}_N$; for finite sequences $A$ and $B$, even in the case of two state Markov chains, the sets of words of length $N$ may not coincide. $A$ and $B$ are sequences

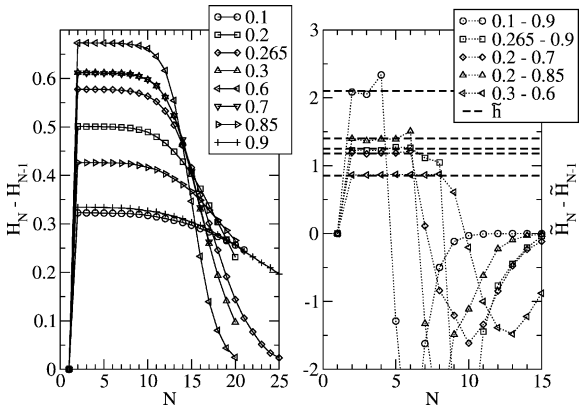


Fig. 2. Left: $H_N - H_{N-1}$ versus $N$ for a Markov process of order 1 with symmetrical transition matrix (see Eq. (25)) calculated numerically using a sequence of 20,000 symbols, for different values of the parameter $w$. The plateau (reached at $N = 2$) corresponds to the theoretical $h$, while the successive decay of the curves is due to poor statistics. Right: cross-entropy $\tilde{h}_N = \tilde{H}_N - \tilde{H}_{N-1}$ for different pairs $(A, B)$ of such Markov processes, characterized by parameters $w_A$, $w_B$. The plateaus (put in evidence by dashed lines) correspond to the theoretical value $\tilde{h}$.

of length 20,000 generated with the symmetric one-step Markov processes with different transition matrices $W$, i.e. with different parameters $w_A$ and $w_B$.

It can be seen that the plateau representing $h$ is reached at $N = 2$, as expected for Markov chains of order $k = 1$. Moreover, the effect of finite size can be seen: the sequences considered are 20,000 symbols long, therefore, invoking the Shannon–McMillan theorem, one has that $N$ must not be too large in order to satisfy the condition that the number of typical $N$-sequences be much smaller than the length of the sequence, i.e. $\mathcal{N} = 2^{hN} \ll 20{,}000$. Otherwise the statistics becomes too poor and $h_N$ rapidly departs from $h$. In the right plot of Fig. 2 we show the behavior of $\tilde{h}_N$: the first plateau of the curves in this graph provides an estimate of the cross-entropy $\tilde{h}(B\|A)$. This figure shows how finite size effects appear in the computation of $d(B\|A)$, well before those appearing in the computation of $h$; this is a direct consequence of the operative definition used in this computation: in order to have a good estimate of $\tilde{h}_N$ a large amount of $N$-sequences common both to $A$ and $B$ is indeed needed, reducing the value of the finite size cut-off. The scaling of $P(L_A, L_B)$ for pairs of Markov sequences is shown in Fig. 3. Again a good

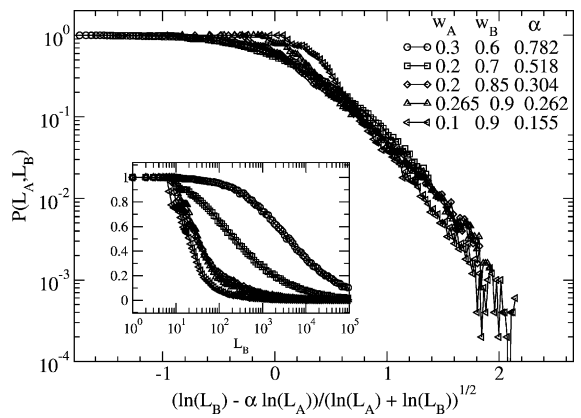


Fig. 3. Collapse of $P(L_A, L_B)$ versus the rescaled coordinate for Markov processes of order 1 and symmetric transition matrix (see Eq. (25)) with different values of the pairs $(w_A, w_B)$, with $L_A = 20{,}000$. In the figure are also indicated the values of $\alpha$ (see (24)). In the inset the data without rescaling.

collapse is obtained using the previously proposed scaling form (24). It is also clear that the collapse fails for pairs of processes with $\alpha \ll 1$, i.e. the pairs with the strongest difference in the transition matrix.

### 5.3. Non-Markovian sequences: Lozi map symbolic dynamics

It is interesting to probe a class of signals (i.e. sequences) with a higher degree of complexity, i.e. large memory and forbidden words. Chaotic dynamical systems are a rather natural source of such nontrivial signals. A symbolic sequence can be associated to the dynamical system by means of a partition of the phase space $\Omega$, i.e. $\{\omega_i\}$ with $m$ elements such that $\bigcup_{i=1}^{m} \omega_i = \Omega$ and $\omega_i \cap \omega_j = 0$ for every $i$ and $j$ in $[1, m]$. Every trajectory $\mathbf{x}(t)$ is therefore mapped into a sequence of symbols of the $m$-alphabet. An interesting nontrivial example can be obtained with a binary partition of the $x$ variable of the Lozi map, defined as

$$x(n+1) = -a|x(n)| + y(n) + 1, \qquad y(n+1) = bx(n), \tag{28}$$

where $a$ and $b$ are parameters. The sequence of symbols used in the following test is obtained taking 0 when $x \leq 0$ and 1 when $x > 0$. For $b = 0.5$, numerical studies show that the Lozi map is chaotic for $a$ in the interval (1.51, 1.7). For a discussion of the Lozi map, computation of Lyapunov exponents and representation of its symbolic dynamics in terms of Markov chains, see [47].

Fig. 4 reports the numerical computation of $H_N$ and $\tilde{H}_N$ (the block entropy and the block cross-entropy) for several sequence lengths, using always the same pair of processes $a_A = 1.56$ and $a_B = 1.52$. The aim is putting in evidence finite size effects as well as estimating Shannon and Kullback–Leibler entropies needed for the collapse of $P(L_A, L_B)$. The estimate of $d(B\|A)$ and $h(B)$ and therefore of $\alpha$ is obtained with a level of confidence of 10%. Due to statistic effects, we measure the slopes of the curves (both for $H_N$ and $\tilde{H}_N$) in the range of $N$ where the slope is constant, as already done in Fig. 2. Let us note that in the symbolic sequence generated by the Lozi map there is also the problem of the lack of equivalence between $\{C_N\}$ and $\{C_N\}^*$ (see Eq. (26)). However one must note that the $A + B$ zipping procedure used in our analysis finds only sequences contained in both $A$ and $B$. It is natural, therefore, to measure $D_N^*$ (which in this case is different from $D_N$) and from this estimate $d^*$.

Fig. 4 is particularly enlightening from the point of view of the meaning of the effective measure complexity $C$ defined in Eq. (7). A naive order 1 Markovian approximation of the map is far from reproducing the dynamical properties of the Lozi map. This can be appreciated in Fig. 4, noting that $C$ is not small.

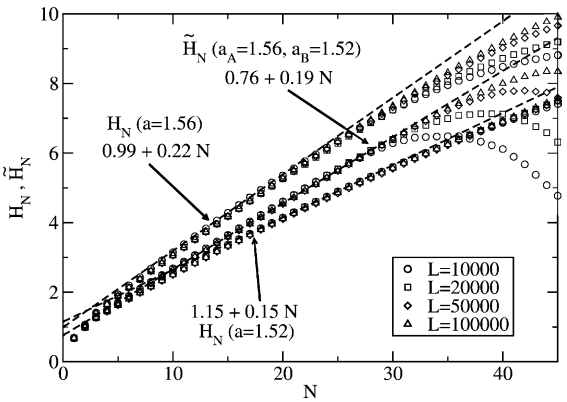


Fig. 4. $H_N$ and $\tilde{H}_N$ versus $N$ for sequences of symbols obtained with a binary partition of the Lozi map. The $H_N$ are calculated using Lozi map with parameter $a = 1.52$ and $a = 1.56$. The $\tilde{H}_N$ are calculated using pairs of Lozi map with $a_A = 1.56$ and $a_B = 1.52$. All calculations have been performed with sequences of different length $L$, to probe finite size effects.

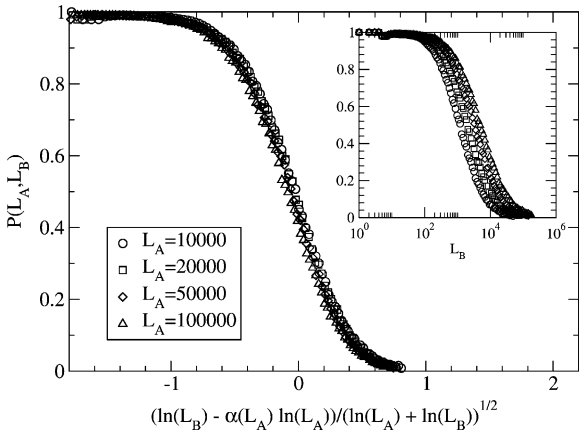


Fig. 5. Collapse of $P(L_A, L_B)$ versus the rescaled coordinate for sequences of symbols obtained with a binary partition of the Lozi map with parameters pairs $(a_A, a_B) = (1.56, 1.52)$, using an estimate of $\alpha = 0.78$ obtained using the values $h(B) = 0.15$ and $\tilde{h} = 0.19$ (see Fig. 4) In the inset the same data are shown versus $L_B$, i.e. without rescaling.

Finally, in Fig. 5 it is shown that the collapse of the learning curves $P(x, y)$ is very well verified, using again averages on the $B$ sequence (i.e. different initial conditions) and different lengths for the $A$ sequence. In this case we have used $d^*(B\|A)$ instead of $d(B\|A)$ to compute $\alpha$, i.e.:

$$\alpha = \frac{h(B)}{h(B) + d^*(B\|A)}. \tag{29}$$

## 6. An experiment of recognition

The last set of results concerns one of the main motivation of this analysis, i.e. its practical applications. The algorithm proposed in [23,25] has its main justification in its efficiency on the framework of sequence recognition: the algorithm is able to provide an estimate of the Kullback–Leibler entropy of a sequence of unknown provenance relatively to a set of sequences whose provenance is certain (known sources) and used as reference sequences, giving the most "similar" sequence and therefore the most probable source for the sequence of unknown provenance. In this context, we have checked that this recipe well recognizes a symbolic sequence drawn from the class of Lozi maps. Though the results are very preliminary and a systematic analysis should be in order, some interesting conclusions can be drawn.

Fig. 6 reports the result of this test. A Lozi map with $a = 1.6$, $b = 0.5$ and initial condition $x = 0.1$, $y = 0.1$ has been used to generate the sequence $A$, of length 10,000, that will be used as unknown sequence. As probing sequences we have generated two sets of sequences, $B$ and $B^*$, respectively, obtained with Lozi maps with the parameters $b = 0.5$ and $a_B = a_{B^*}$ varying between 1.52 and 1.7. The sequences $B$ has length of 10,000 while sequence $B^*$ has length of 5000 or 1000. The quantities plotted in the inset are the lengths of the compressed code (with the LZ77 algorithm, see the discussion in Section 2), i.e. $C(X)$ is the length of the code obtained by compressing the sequence $X$. Data relative to the compression of the sequences $B + B^*$ and $A + B^*$ have been obtained by averaging over 100 different choices of initial conditions. The quantity computed and reported in the main graph is an estimate of the Kullback–Leibler entropy $d(B\|A)$, as a difference (per bit) between $C(A + B^*) - C(A)$ and $C(B + B^*) - C(B)$ which are the estimates of the block cross-entropy and of the entropy of B, respectively. The bottom plot shows

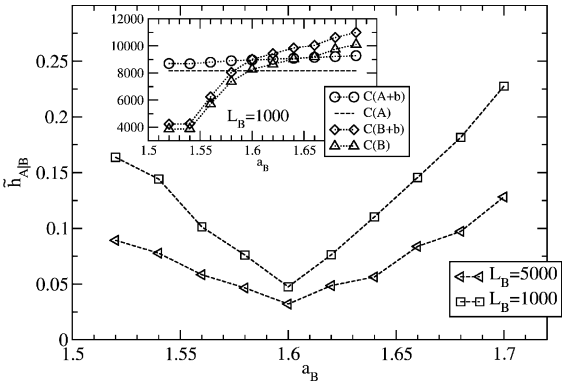


Fig. 6. Estimate, by means of LZ77 compression, of $D(B\|A)$ (see text) of the Kullback–Leibler entropy relative to different pairs $(A, B)$ of sequences of symbols: each pair is composed by a fixed sequence $A$ obtained as a binary partition of a Lozi map with parameter $a_A = 1.6$ and a variable sequence $B$ obtained either as a binary partition of a Lozi map with variable parameter $a_B$. The sequences have length $L_A = L_B = 10{,}000$. The estimate of the Kullback–Leibler entropy has its minimum in correspondence of the pair $(A, A)$ (i.e. when $B$ comes from a Lozi map with $a_B = a_A$): this indicates that this estimate of $D(B\|A)$ is capable of recognizing in the space of Lozi maps. In the inset the lengths of the LZ-compressed sequences are reported, where $B^*$ is always a sequence of the same kind of $B$ (note that $(L_A)^\alpha \simeq 1300$ and therefore $L_B^* = 1000$, and $L_B^* = 5000$ are below and beyond the cross-over threshold, respectively).

very well how this simple recipe leads to a perfect recognition of the correct value of $a = 1.6$: the estimate of the Kullback–Leibler entropy has in fact an absolute minimum for that value.

In Fig. 6 one can also appreciate the usefulness of the theoretical analysis of Section 4, i.e. the fact that $(L_A)^\alpha$ is a good estimate of the best length $L_B$ of the probe sequences $B$ to obtain the optimal resolution in the recognition process. In fact in Section 4 we conjectured (and successively verified with numerical experiments) that when $L_B$ is smaller than the cross-over length $L_A^\alpha$, the LZ77 algorithm is encoding the sequence $B$ with the "language" of $A$ and therefore the length of the encoded sequence is effectively a measure of the distance between the two languages. Using the previous value $\alpha = 0.78$ as a rough estimate for every other choice of the map parameter $a$, and given $L_A = 10{,}000$, one obtains for the cross-over length $\sim 1300$. In the figure, the resolution power of the LZ77 algorithm with $L_B = 1000$ is much higher than that with $L_B = 5000$.

## 7. Conclusions

We have studied the properties of standard sequential compression algorithms in the problem of information extraction from sequences of characters. We have in particular analyzed the learning process that these algorithm perform when they are used to compress heterogeneous data, i.e. data coming from different sources.

The typical benchmark for this study is a finite sequence of $L_A + L_B$ symbols obtained appending a sequence of $L_B$ symbols emitted by a source $B$ to a sequence of $L_A$ symbols emitted by a source $A$. An algorithm like LZ77 [30], after having processed the $A$ part of the sequence, starts encoding the $B$ part using the knowledge acquired while zipping the $A$ part; after a transient the compression algorithm starts encoding the $B$ part using the knowledge coming only from the $B$ part already processed (i.e. the zipper starts learning the $B$ part). We have made a scaling hypothesis that characterizes this transient process in terms of the entropy of the source $B$ and the Kullback–Leibler divergence between the two sequences.

We have studied the finite size scaling (i.e. incorporating fluctuations due to the finite size of the sequences under investigation) by means of numerical experiments on three sets of data coming from different sources: the Bernoulli

scheme, the Markov chain of first order (with symmetric transition matrix) and the symbolic dynamics obtained with a binary partition of the Lozi map. These three examples feature an increasing complexity: the Bernoulli scheme emits sequences of uncorrelated random symbols; the Markov chain of first order is the simplest way to enforce correlations among symbols in the sequences; finally the Lozi map has the property of having an higher effective measure complexity [33,34]. The scaling hypothesis is very well verified in all the cases investigated, pointing out the generality of the result.

These results have a practical importance in the analysis of a recently proposed scheme that computes the informational remoteness between two sequences [25]: in fact this scheme employs a variant of the LZ77 algorithm and gives the best estimate of the remoteness (Kullback–Leibler divergence) when the length of the second sequence is chosen of the order of the threshold value of the learning function we have introduced in this work. We have investigated quantitatively this point, showing that the resolution power of the recognition scheme proposed in [25] is highly improved when the length of the second sequence is chosen according to the analysis of the transient. Sequences too short or too long can give bad estimates of the Kullback–Leibler divergence and therefore a big uncertainty in the recognition of similar sequences.

Another important field of application is that of the segmentation of heterogeneous sequences, i.e. the identification of the boundaries between regions featuring very different properties which, depending on the sequences considered, can correspond to very different phenomena (catastrophic events in geophysical time series, or boundaries between different sections in genetic sequences just to quote a couple of examples). In all these cases one could try to exploit the features of data compression techniques at the interface between heterogeneous regions in order to define and optimize suitable observables sensitive to sudden changes.

## Acknowledgements

VL, AP and AV acknowledge support from the INFM *Center for Statistical Mechanics and Complexity* (SMC) and MIUR (Cofin 2001023848_008).

## References

[1] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, Cambridge Nonlinear Science Series, Cambridge University Press, Cambridge, 1997.
[2] H.D.I. Abarbanel, Analysis of Observed Chaotic Data, Springer, New York, 1996.
[3] R. Badii, A. Politi, Complexity, Hierarchical Structures and Scaling in Physics, Cambridge University Press, Cambridge, 1997.
[4] C.E. Shannon, The Bell Syst. Tech. J. 27 (1948) 623.
[5] W.H. Zurek (Ed.), Complexity, Entropy and Physics of Information, Addison-Wesley, Redwood City, 1990.
[6] A.N. Kolmogorov, Prob. Inform. Trans. 1 (1965) 1.
[7] G.J. Chaitin, J. Assoc. Comp. Mach. 13 (1966) 547.
[8] G.J. Chaitin, Information, Randomness and Incompleteness, World Scientific, Singapore, 1990.
[9] R.J. Solomonoff, Inform. Contr. 7 (1964) 1 and 224.
[10] M. Li, P. Vitányi, An Introduction to Kolmogorov Complexity and Its Applications, Springer, Berlin, 1997.
[11] J. Ziv, N. Merhav, IEEE Trans. Inform. Theory 39 (1993) 1270.
[12] A. Milosavljević, Mach. Learn. 21 (1995) 35–50.
[13] M. Farach, M. Noordewier, S. Savari, L. Shepp, A. Wyner, J. Ziv, in: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, January 22–24, 1995, pp. 48–57.
[14] S. Verdú, Fifty years of Shannon theory, IEEE Trans. Inform. Theory 44 (1998) 2057.
[15] D. Lind, B. Marcus, Symbolic Dynamic and Coding, Cambridge University Press, Cambridge, 1995.
[16] V. Benci, C. Bonanno, S. Galatolo, G. Menconi, M. Virgilio, Preprint cond-mat/0210654, and references therein.
[17] G. Boffetta, M. Cencini, M. Falcioni, A. Vulpiani, Phys. Rep. 356 (2002) 367.
[18] T.C. Bell, J.C. Cleary, I.H. Witten, Text Compression, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[19] W.J. Teahan, in: Proceedings of the International Conference on Content-based Multimedia Information Access (RIAO 2000), CID-CASIS, Paris, 2000, pp. 943–961.
[20] P. Juola, in: Proceedings of the New Methods in Language Processing 3, Sydney, 1998.
[21] R. El-Yaniv, S. Fine, N. Tishby, Adv. Neural Inform. Process. Syst. 10 (1997) 465–471.
[22] N. Thaper, MS in Computer Science, Masters Thesis, MIT Press, Cambridge, MA, 2001.
[23] O.V. Kukushkina, A.A. Polikarpov, D.V. Khmelev, Problemy Peredachi Inform. 37 (2000) 96–108 (in Russian). Translated in English in Problems Inform. Trans. 37 (2001) 172–184.
[24] M. Li, Private communication.
[25] D. Benedetto, E. Caglioti, V. Loreto, Phys. Rev. Lett. 88 (2002) 048702.
[26] M. Li, J. Badger, X. Chen, S. Kwong, P. Kearney, H. Zhang, Bioinformatics 17 (2001) 149–154.
[27] S. Grumbach, F. Tahi, Inform. Process. Manage. 30 (1994) 875–886.
[28] D. Loewenstern, et al., DIMACS Technical Report 95-04.
[29] http://www.gzip.org/.
[30] A. Lempel, J. Ziv, IEEE Trans. Inform. Theory 23 (1977) 337–343.
[31] A.I. Khinchin, Mathematical Foundations of Information Theory, Dover, New York, 1957.
[32] D. Welsh, Codes and Cryptography, Clarendon Press, Oxford, 1989.
[33] J.P. Crutchfield, N.H. Packard, Int. J. Theor. Phys. 21 (1982) 433–466;
J.P. Crutchfield, N.H. Packard, Physica D 7 (1983) 201–223.
[34] P. Grassberger, Int. J. Theor. Phys. 25 (1986) 907–938;
P. Grassberger, in: H. Atmanspacher, H. Scheingraber (Eds.), Large Information Dynamics, Plenum Press, New York, 1991, pp. 15–33.
[35] J.P. Crutchfield, D.P. Feldman, cond-mat/0102181.
[36] A.D. Wyner, J. Ziv, Proc. IEEE 82 (1994) 872–877.
[37] A. Savari, IEEE Trans. Inform. Theory 44 (1998) 787–791.
[38] J.R. Pierce, Introduction to Information Theory: Symbols, Signals and Noise, Dover, New York, 1980.
[39] P. Grassberger, Data compression and entropy estimates by non-sequential recursive pair substitution, preprint archive: physics/0207023.
[40] S. Kullback, R.A. Leibler, Ann. Math. Statist. 22 (1951) 79–86.
[41] S. Kullback, Information Theory and Statistics, Wiley, New York. 1959.
[42] T. Cover, J. Thomas, Elements of Information Theory, Wiley, New York, 1991.
[43] S.R.S. Varadhan, Large Deviations and Applications, SIAM, Philadelphia, 1984.
[44] A.J. Wyner, IEEE Trans. Inform. Theory 43 (5) (1997) 1452–1464.
[45] D. Aldous, P. Shields, Prob. Theory Related Fields 79 (1988) 509–542.
[46] P. Jacquet, W. Szpankowski, J. Tang, Algorithmica 31 (2001) 318–360.
[47] A. Crisanti, G. Paladin, A. Vulpiani, Products of Random Matrices in Statistical Physics, Springer, Berlin, 1993.